3Б

УДК 681.324

А.А. Баркалов[1], С.А. Ковалев[2], Р.М. Бабаков[3], Д.В. Николаенко[2]
[1] Университет Зеленогурский, г. Зеленая Гура, Польша
[2] Донецкий национальный технический университет, Украина
[3] Государственный университет информатики и искусственного интеллекта, Украина

Метод синтеза композиционного микропрограммного устройства управления с разделением кодов и кэшированием

Разработан метод синтеза базовой структуры композиционного микропрограммного устройства управления с разделением кодов и кэшированием микрокоманд. В основе метода лежит эвристическое распределение операторных линейных цепей в адресном пространстве управляющей памяти, обеспечивающее максимальное для реализуемого алгоритма управления значение вероятности кэш-попаданий. Приведен пример использования разработанного метода синтеза.

Общая постановка проблемы

Важным составным элементом современных цифровых вычислительных систем является устройство управления (УУ), одним из способов реализации которого является композиционное микропрограммное устройство управления (КМУУ) с разделением кодов, характеристики которого во многом определяют характеристики системы в целом [1]. В работе [2] для увеличения быстродействия схемы базовой структуры КМУУ с разделением кодов предлагается использовать принцип кэширования сигналов, что требует модификации исходной структуры путем введения дополнительного блока, реализующего функции кэш-памяти. Использование данного модуля позволяет снизить среднее время доступа к относительно медленным блокам, реализованным в базисе ПЗУ или ППЗУ, что приводит к уменьшению средней длительности такта работы устройства и к увеличению его среднего быстродействия. Условимся обозначать полученную в [2] базовую структуру КМУУ с разделением кодов и кэшированием через S_{1k}.

Характерной особенностью всех структур КМУУ является их аппаратная привязка к реализуемому алгоритму управления. Данная особенность позволяет провести оптимизацию схемы КМУУ для каждого конкретного случая реализации. В работе [3] предложен эвристический метод оптимизации эффективности структуры S_{1k}, основанный на специальной адресации микрокоманд в управляющей памяти.

В настоящей статье предлагается метод синтеза логической схемы базовой структуры КМУУ с разделением кодов и кэш-памятью, основанный на предложенном в [3] эвристическом алгоритме адресации микрокоманд.

Метод синтеза КМУУ с разделением кодов и кэш-памятью микрокоманд

Исходными данными при синтезе структуры S_{1k} являются:
– исходная граф-схема алгоритма (ГСА);
– вероятности выполнения логических условий;
– тип архитектуры и размеры блока данных модуля кэш-памяти.

Для синтеза структуры S_{1k} предлагается метод, включающий следующие основные этапы:

1. Разбиение множества микрокоманд заданной ГСА на операторные линейные цепи.
2. Получение с использованием разработанного эвристического алгоритма шести значений вероятности кэш-попаданий, обусловленных шестью стратегиями выбора нераспределенных операторных линейных цепей (ОЛЦ), и выбор варианта размещения ОЛЦ с максимальным значением вероятности кэш-попаданий.
3. Кодирование ОЛЦ и адресация микрокоманд согласно выбранному варианту размещения.
4. Формирование содержимого ПЗУ управляющей памяти.
5. Построение прямой структурной таблицы.
6. Формирование системы функций переходов.
7. Синтез функциональной схемы устройства в заданном элементном базисе.

Пример синтеза базовой структуры КМУУ с разделением кодов и кэш-памятью микрокоманд

Рассмотрим пример синтеза структуры S_{1k} для ГСА G_1 (рис. 1). Содержимое операторных вершин ГСА представлено в табл. 1.

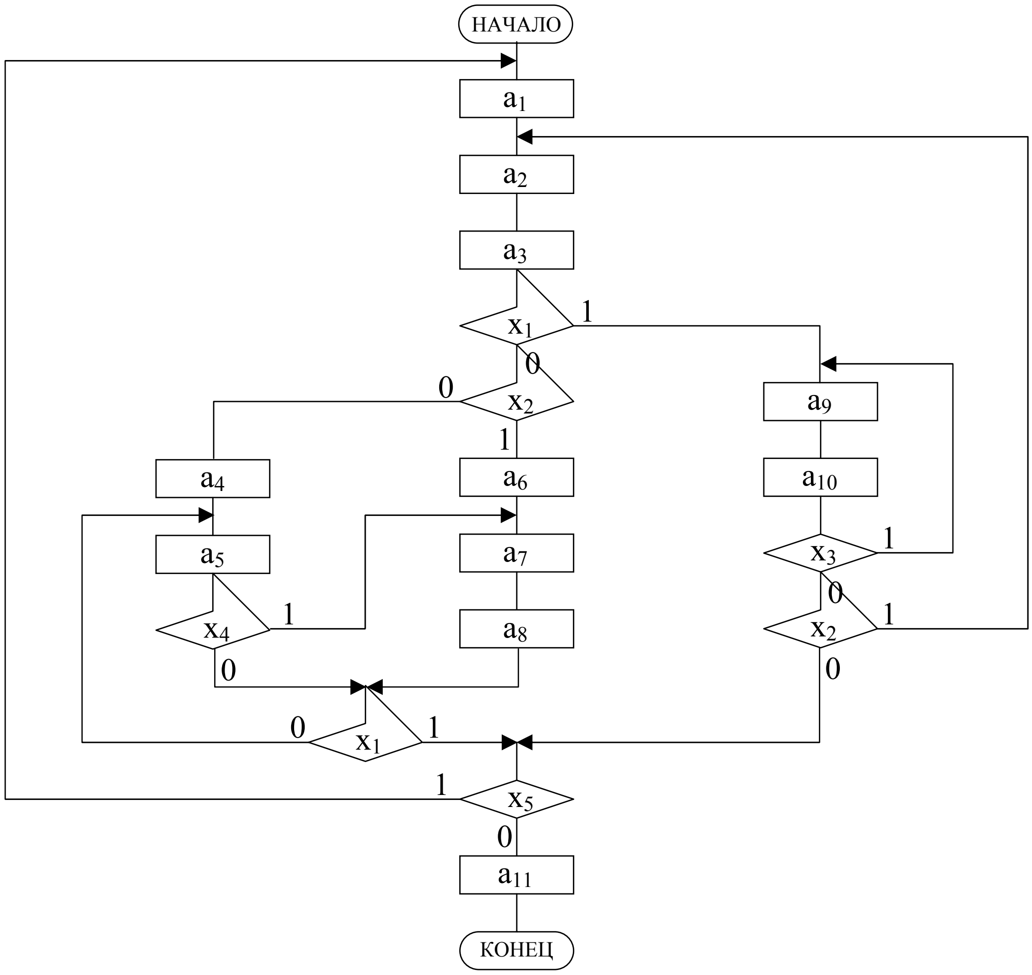


Рисунок 1 – Граф-схема алгоритма G

Отметим, что для обеспечения функционирования КМУУ в микрокоманды, являющиеся выходами ОЛЦ, добавлена микрооперация y_0, а в МК a_{11}, являющейся последней в ГСА, добавлена микрооперация y_k.

Пусть известны (на основании теоретических или экспериментальных исследований) вероятности переходов: $p(x_1) = 0,2$; $p(x_2) = 0,7$; $p(x_3) = 0,5$; $p(x_4) = 0,1$; $p(x_5) = 0,9$.

Таблица 1 – Содержимое операторных вершин ГСА G_1

a_i	$Y(a_i)$	a_i	$Y(a_i)$
a_1	$y_1\ y_2\ y_3$	a_7	$y_1\ y_4$
a_2	y_4	a_8	$y_2\ y_5\ y_0$
a_3	$y_2\ y_5\ y_0$	a_9	y_3
a_4	$y_1\ y_3$	a_{10}	$y_2\ y_0$
a_5	$y_3\ y_0$	a_{11}	$y_1\ y_2\ y_4\ y_k$
a_6	$y_2\ y_3\ y_5$		

1. Сформируем ОЛЦ известным способом [1]: $O_1 = \{a_1, a_2, a_3\}$, $O_2 = \{a_4, a_5\}$, $O_3 = \{a_6, a_7, a_8\}$, $O_4 = \{a_9, a_{10}\}$, $O_5 = \{a_{11}\}$. Максимальный размер ОЛЦ $N_{max} = 3$, следовательно, $R(T) =]\log_2(N_{max})[= 2$. Для пяти ОЛЦ $R(\tau) =]\log_2 5[= 3$. Разрядность адреса микрокоманды $R = R(T) + R(\tau) = 5$, емкость ПЗУ схемы управляющей памяти (УП) составит $2^R = 32$ слова. При размере строки кэша $N_C = 8$ слов адресное пространство УП делится на четыре блока: $B_1 – B_4$. Внутри каждой ОЛЦ микрокоманды имеют последовательные номера, начиная с нулевого.

Определим входы ОЛЦ I_j^i, где i – номер входа, j – номер ОЛЦ:

$$I_1^1 = a_1;\ I_2^1 = a_2;\ I_2^1 = a_4;\ I_2^2 = a_5;\ I_3^1 = a_6;\ I_3^2 = a_7;\ I_4^1 = a_9;\ I_5^1 = a_{11}.$$

2. В табл. 2 показаны экспериментальные значения вероятности кэш-попаданий для различных стратегий [3], полученные с помощью специально разработанной программной имитационно-аналитической модели рассматриваемой структуры. Определение значений вероятности кэш-попаданий выполнено для кэш-памяти, блок данных которой имеет одну строку размером в 8 слов. Это исключает влияние на результат типа архитектуры кэш-памяти.

Таблица 2 – Значения p_h для различных стратегий

Стратегия	p_h	Стратегия	p_h
1	0,8273	4	0,7877
2	0,7877	5	0,7877
3	0,8273	6	0,7877

Сходность значений p_h для различных стратегий объясняется относительно низкой сложностью исходной ГСА.

Очевидно, что стратегии 1 и 3 дают максимальные значения вероятности кэш-попаданий. В качестве варианта размещения ОЛЦ выберем результат для стратегии 3. Для данной стратегии размещение ОЛЦ в УП выполнено в работе [3] и в окончательном виде отражено в табл. 3.

Таблица 3 – Адресация ОЛЦ после оптимизации (G_1)

Адрес	ОЛЦ	Адрес	ОЛЦ
0	O_2	12	O_4
4	O_3	16	O_5
8	O_1		

3. С учетом выбранного варианта размещения кодирование ОЛЦ представлено в табл. 4, а адресация микрокоманд – в табл. 5.

Таблица 4 – Кодирование ОЛЦ

O_i	$K(O_i)$ $\tau_1\ \tau_2\ \tau_3$	O_i	$K(O_i)$ $\tau_1\ \tau_2\ \tau_3$
O_1	0 1 0	O_4	0 1 1
O_2	0 0 0	O_5	1 0 0
O_3	0 0 1		

Таблица 5 – Адресация микрокоманд

a_i	$\tau_1\ \tau_2\ \tau_3\ T_1\ T_2$	a_i	$\tau_1\ \tau_2\ \tau_3\ T_1\ T_2$
a_1	0 1 0 0 0	a_7	0 0 1 0 1
a_2	0 1 0 0 1	a_8	0 0 1 1 0
a_3	0 1 0 1 1	a_9	0 1 1 0 0
a_4	0 0 0 0 0	a_{10}	0 1 1 0 1
a_5	0 0 0 0 1	a_{11}	1 0 0 0 0
a_6	0 0 1 0 0		

4. Содержимое ПЗУ управляющей памяти строится на основании табл. 1 и 5 и представлено в табл. 6.

Таблица 6 – Содержимое ПЗУ управляющей памяти

Адрес $\tau_1\ \tau_2\ \tau_3\ T_1\ T_2$	Микрооперации $y_1\ y_2\ y_3\ y_4\ y_5\ y_0\ y_k$	Адрес $\tau_1\ \tau_2\ \tau_3\ T_1\ T_2$	Микрооперации $y_1\ y_2\ y_3\ y_4\ y_5\ y_0\ y_k$
0 0 0 0 0	1 0 1 0 0 0 0	1 0 0 0 0	1 1 0 1 0 0 1
0 0 0 0 1	0 0 1 0 0 1 0	1 0 0 0 1	* * * * * * *
0 0 0 1 0	* * * * * * *	1 0 0 1 0	* * * * * * *
0 0 0 1 1	* * * * * * *	1 0 0 1 1	* * * * * * *
0 0 1 0 0	0 1 1 0 1 0 0	1 0 1 0 0	* * * * * * *
0 0 1 0 1	1 0 0 1 0 0 0	1 0 1 0 1	* * * * * * *
0 0 1 1 0	0 1 0 0 1 1 0	1 0 1 1 0	* * * * * * *
0 0 1 1 1	* * * * * * *	1 0 1 1 1	* * * * * * *
0 1 0 0 0	1 1 1 0 0 0 0	1 1 0 0 0	* * * * * * *
0 1 0 0 1	0 0 0 1 0 0 0	1 1 0 0 1	* * * * * * *
0 1 0 1 0	0 1 0 0 1 1 0	1 1 0 1 0	* * * * * * *
0 1 0 1 1	* * * * * * *	1 1 0 1 1	* * * * * * *
0 1 1 0 0	0 0 1 0 0 0 0	1 1 1 0 0	* * * * * * *
0 1 1 0 1	0 1 0 0 0 1 0	1 1 1 0 1	* * * * * * *
0 1 1 1 0	* * * * * * *	1 1 1 1 0	* * * * * * *
0 1 1 1 1	* * * * * * *	1 1 1 1 1	* * * * * * *

Можно констатировать, что при емкости ПЗУ 32 слова используются лишь 11 адресов, то есть 34,4 % емкости УП. Данный факт обусловлен особенностями принципа разделения кодов [1].

5. После того, как известны адреса микрокоманд (табл. 5), прямая структурная таблица строится известным способом [1] и для нашего примера представлена в табл. 7. Таблица включает следующие столбцы:

O_i – текущая ОЛЦ;

$K(O_i)$ – код текущей ОЛЦ;

I_j^i – i-й вход ОЛЦ O_j;

$K(I_j^i)$ – полный адрес МК, являющейся i-м входом ОЛЦ O_j;

X_h – логические условия, обеспечивающие данный переход;

Φ_h – функции возбуждения регистра памяти;

Ψ_h – функции возбуждения счетчика адреса;

h – номер перехода.

При построении ПСТ предполагается использование в качестве элементов памяти триггеров D-типа.

Таблица 7 – Прямая структурная таблица КМУУ S_{1k} (G_1)

O_i	$K(O_i)$	I_j^i	$K(I_j^i)$	X_h	Φ_h	Ψ_h	h
O_1	0 1 0	I_2^1	0 0 0 0 0	$\bar{x}_1\bar{x}_2$	–	–	1
		I_3^1	0 0 1 0 0	$\bar{x}_1 x_2$	D_3	–	2
		I_4^1	0 1 1 0 0	x_1	D_2 D_3	–	3
O_2	0 0 0	I_3^2	0 0 1 0 1	x_4	D_3	D_5	4
		I_2^2	0 0 0 0 1	$\bar{x}_4\bar{x}_1$	–	D_5	5
		I_1^1	0 1 0 0 0	$\bar{x}_4 x_1 x_5$	D_2	–	6
		I_5^1	1 0 0 0 0	$\bar{x}_4 x_1 \bar{x}_5$	D_1	–	7
O_3	0 0 1	I_2^2	0 0 0 0 1	$\bar{x}_1$	–	D_5	8
		I_1^1	0 1 0 0 0	$x_1 x_5$	D_2	–	9
		I_5^1	1 0 0 0 0	$x_1 \bar{x}_5$	D_1	–	10
O_4	0 1 1	I_4^1	0 1 1 0 0	x_3	D_2 D_3	–	11
		I_2^1	0 1 0 0 1	$\bar{x}_3 x_2$	D_2	D_5	12
		I_1^1	0 1 0 0 0	$\bar{x}_3\bar{x}_2 x_5$	D_2	–	13
		I_5^1	1 0 0 0 0	$\bar{x}_3\bar{x}_2\bar{x}_5$	D_1	–	14

6. Система функций переходов образовывается функциями D_1 – D_3, отождествляемыми с кодом ОЛЦ $\tau_1\tau_2\tau_3$, и функциями D_4, D_5, отождествляемыми с кодом МК внутри ОЛЦ T_1T_2.

Функции строятся по известной методике [1] на основании содержимого ПСТ (табл. 7). Например: $D_3 = \bar{\tau}_1 \tau_2 \bar{\tau}_3 \bar{x}_1 x_2 \vee \bar{\tau}_1 \tau_2 \bar{\tau}_3 x_1 \vee \bar{\tau}_1 \bar{\tau}_2 \bar{\tau}_3 x_4 \vee \bar{\tau}_1 \tau_2 \tau_3 x_3$.

С целью уменьшения аппаратурных затрат в схеме устройства может быть выполнена минимизация функций переходов.

7. Построение функциональной схемы КМУУ S_{1k} выполняется традиционно и рассмотрено в [1].

Заключение

Предложенный в статье метод синтеза базовой структуры композиционного микропрограммного устройства управления с разделением кодов и кэш-памятью позволяет эвристически выбрать оптимальное размещение ОЛЦ реализуемой ГСА в адресном пространстве ПЗУ управляющей памяти, дающее максимальную величину вероятности кэш-попаданий алгоритма. Проведенные авторами эксперименты показали, что разработанный метод на всех этапах поддается алгоритмизации и может быть в дальнейшем использован в системах автоматизированного проектирования цифровых устройств управления.

Литература

1. Баркалов А.А. Синтез устройств управления на программируемых логических устройствах. – Донецк: ДонНТУ, 2002. – 262 с.
2. Баркалов А.А., Ковалев С.А., Бабаков Р.М., Николаенко Д.В. Организация композиционных микропрограммных устройств управления с разделением кодов и кэш-памятью // Искусственный интеллект. – 2007. – № 3. – С. 135-138.
3. Баркалов А.А., Ковалев С.А., Бабаков Р.М., Николаенко Д.В. Эвристический алгоритм оптимизации размещения микрокоманд в композиционном микропрограммном устройстве управления с разделением кодов и кэш-памятью // Искусственный интеллект. – 2008. – № 1. – С. 20-29.

О.О. Баркалов, С.О. Ковальов, Р.М. Бабаков, Д.В. Ніколаєнко

Метод синтезу композиційного мікропрограмного пристрою керування з розподілом кодів і кешуванням

Розроблено метод синтезу базової структури композиційного мікропрограмного пристрою керування з розподілом кодів і кешуванням мікрокоманд. В основі методу лежить евристичний розподіл операторних лінійних ланцюгів в адресному просторі керуючої пам'яті, що забезпечує максимальне для реалізованого алгоритму керування значення ймовірності кеш-влучень. Наведено приклад використання розробленого методу синтезу.

A.A. Barkalov, S.A. Kovalev, R.M. Babakov, D.V. Nikolaenko

The method of Synthesis of Compositional Microprogram Control Unit with Division of Codes and Caching

The method of synthesis of base structure of the compositional microprogram control unit with division of codes and caching of microinstructions is developed. It is based on heuristic distribution of operational linear circuits in address space of the managing memory, providing maximal for realized control algorithm value of probability of cache-hits. The example of using of the developed method of synthesis is resulted.

Статья поступила в редакцию 17.04.2008.